IN THE SPECIFICATION:

Kindly amend the specification as indicated below.

[0076] Once the query system 132 receives the first result set and the second result set, the query system 132 combines the two result sets 322. The query system may combine the result sets in a number of ways. For example, in one embodiment of the present invention, the query system merges the two result sets, the results being ranked by the relevancy score regardless of from which query a particular result originated. In another embodiment of the present invention, the query system 132 takes into account the search query that was executed in determining how to combine the result sets. For example, the query system 132 may rate results acquired from a local index higher than those results acquired from a global index or vice versa. In such an embodiment, the result sets that are shown to the user will have the local results listed above the global results. Embodiments of the present invention may perform additional combining and/or ranking steps here as well. For example in one embodiment of the present invention the query system 132 evaluates the list of article identifiers that are returned from both result sets and eliminates any duplicates. In yet another embodiment, the query system 132 combines the article identifiers from each of the two result sets and then performs additional ranking, weighting, and sorting procedures on each of the result sets. For example, an article identifier that appears in multiple result sets may receive a higher weighting than if it had appeared in only one result set. In one embodiment, the result sets are not merged; they are displayed on one page in separate lists. Once the query system 132 has combined the result sets from the two search queries, the query system 132 transmits the combined result sets to the display processor (128) [[224]] 324.

[0078] Figure 4 is a flowchart illustrating a method for ranking a combined results set in one embodiment of the present invention. In the embodiment shown, the query system 132 receives a first result set at [[202]] 402. For example, the query system extracts a keyword from the buffer of the application the user is currently using and submits the search query to a search engine such as local search engine 122, and in response, the search engine executes a query and returns a result set to the query system 132. The query system 132 then ranks the first result set based on relevancy scores related to the user's current context state [[204]] 404. The query system 132 receives a second result set [[206]] 406. For example, the query system may submit a query to a global search engine 170, such as Google™ Search Engine. The query system 132 receives the second result set [[206]] 406. The query system next ranks the second result set [[208]] 408. In one embodiment of the present invention, the local and global result sets are ranked based on the same set of criteria. In other embodiments, the result sets originating from local and global indexes may be ranked differently based on user specified or other criteria. Those criteria may include click-through data from the user.

[0079] The query system 132 attempts to create a combined result set. The query system 132 may perform this in a number of ways. For example, in the embodiment shown in Figure 4, the query system identifies an article identifier that appears in both the first and second result sets [[210]] 410. An article identifier that appears in both the first and second result sets may be more likely to be of interest to the user than an article identifier that appears in only one of the result sets. The query system 132 creates a combined result set and adds the article identifier that was identified as being in both the first and second result sets to the combined result set [[212]] 412. The query system 132 may repeat the steps ~~210, 212~~ 410, 412 to add additional article identifiers to the combined result sets. The query system

132 may perform additional types of methods in order to add article identifiers to the combined result set. For example, the query system 132 may extract the top ten article identifiers from each of the first result set and the second result set and add those article identifiers to the combined result set. The query system may further eliminate duplicates from the combined result set or perform other operations that are useful in creating a relevant combined result set for the user. Once the query system 132 has created a combined result set with the relevant article identifiers, the query system 132 transmits the combined result set to display processor [[214]] 414. The display processor will then display the combined result set to the user.

[0081] Figure 5 is a flowchart illustrating the method for processing a query in one embodiment of the present invention. In the embodiment shown, the query system 132 first receives the query string 502. The query system 132 may receive the query string in a number of ways. For example, in one embodiment, the query system 132 receives an API call with a query string created via one of the two methods described herein. The query system 132 then sends the raw string to the full text index 504. In response, the query system 132 receives a list of ranked event IDS 506.

[0082] Using the list of ranked event IDS, the query system 132 retrieves the result set at [[208]] 508. For example, in one embodiment of the present invention, the query system 132 iterates over the list of events until enough results are retrieved to provide a query that will result in a result set sufficient for display to a user. During the process of iteration, the query system 132 retrieves the event record from the data store 140, which includes

information such as the type of event and location (e.g., the URL, path, or other location attribute).

[0086] In one embodiment of the present invention, the query system 132 generates snippets to be displayed with or in place of a link and title 510. In one such embodiment, when the query system 132 indexes a web page document, e.g., when the user loads the document, the query system 132 retrieves document context, including, for example, the first one hundred kilobytes of document text. The query system 132 scans the document context for script tags. It removes the script begin tag, the script end tag, and all text in between the two tags. It then does the same for the style tags, removing the style begin tag, the style end tag, and all text between those two tags. Next, the query system 132 removes other tags that are within less than and greater than (<>) symbols. As the query system 132 iterates over the words in the text, the query system 132 builds a list of positions where the search terms are located within the document. The query system stores the information in the data store.

[0087] Once the query system 132 has retrieved the results and created a snippet for each of the results in the result set, the query system 132 reorders results based on information from the document records 512, such as the frequency of access. In other embodiments, other methods are used to reorder the result set. In one embodiment of the present invention, the query system 132 may also store a filter list - details regarding which filters were applied and/or what the results of filtering were - in memory. By storing the filter list in memory, the query system 132 is able to subsequently access the list to determine relevant results for the user's context state and for other purposes, e.g., in one embodiment

the query system 132 accesses the filter list or the memory in order to determine a relevancy score for results appearing in newly retrieved result sets.

[00100] Figure 7 is a flowchart illustrating a method for altering a relevance score in one embodiment of the present invention. In the embodiment shown, the query system 132 receives at least two result sets 702. The two result sets are received in response to the issuance of two queries, for example, the noun-related and verb-related queries discussed in relation to Figure 6 above. The query system 132 merges the result sets to create a merged result set 704. For example, the result sets may include article identifiers that are common to the result sets. If so, the query system 132 eliminates the duplicates when creating the merged result set. The query system 132 transmits the merged result set to the display processor 128 706.

[00111] The query system 132 subsequently receives a search query 9 10. The query system 132 transmits the search query to a search engine. 912. The search engine returns a result set in response to the search query, which the query system 132 receives 914. The result set comprises one or more article identifiers. The article identifiers are associated with a relevance score. The query system 132 uses the click-through data stored in the data store 140 to modify the relevancy score of each of the articles identifiers in the result set based on the type of content, the source of the content, and the keywords used in the search query 916.

[00126] The query system 132 receives the search result set from the search engine 1112. The result set comprises one or more article identifiers. The article identifiers may be sorted in any of a number of ways. The query system 132 uses a second attribute from the user profile to rank the results [[1014]] 1114. For example, the user 112a may prefer results

from a particular website over results provided from any other source. If the user 112a has such a preference, the query system sorts the result set based on that preference.